Mauricio Alfonso 65/09

1. map foldr

Sea $\cancel{map}$ $\dfrac{}{\Gamma \triangleright map : (\sigma \to \tau) \to [\sigma] \to [\tau]}$ T-Map

y $\cancel{foldr}$ $\dfrac{}{\Gamma \triangleright foldr : (\sigma \to \tau \to \tau) \to \tau \to [\sigma] \to \tau}$ T-Foldr

$$\dfrac{}{\underbrace{\emptyset \triangleright map : ((a \to b \to b) \to (b \to [a] \to b)) \to [a \to b \to b] \to [b \to [a] \to b]}_{(*1)}}$$

Por T-map tomando
$\Gamma = \emptyset$
$\sigma = a \to b \to b$
$\tau = b \to [a] \to b$

$$\dfrac{}{\underbrace{\emptyset \triangleright foldr : (a \to b \to b) \to b \to [a] \to b}_{(*2)}}$$

Por T-Foldr tomando $\Gamma = \emptyset$ $\sigma = a$ $\tau = b$

$$\dfrac{(*1) \qquad (*2)}{\emptyset \triangleright map\ foldr : [a \to b \to b] \to [b \to [a] \to b]}$$

T-App Tomando
$\Gamma = \emptyset$
$\sigma = (a \to b \to b) \to (b \to [a] \to b)$
$\tau = [a \to b \to b] \to [b \to [a] \to b]$

Luego map foldr es tipable.

Mauricio Alfonso 65/09

2. a. Asumo que el tipo id representa infinitos tipos algebraicos posibles.

Defino Δ contexto de declaraciones un conjunto de expresiones del tipo lettipe id $= C_1\sigma_{1,1} \cdots \sigma_{1K_1}; \cdots ; C_n\sigma_{n,1} \cdots \sigma_{n\,K_n}$.

Por ejemplo {lettype maybeInt = Just int ; Nothing} es un Δ válido.

$$\frac{\Gamma ; \Delta \cup \{\text{lettype id} = C_1\sigma_{1,1}\cdots\sigma_{1K_1}; \cdots ; C_n\sigma_{n,1}\cdots\sigma_{nK_n}\} \triangleright e : \tau}{\Gamma ; \Delta \triangleright \text{lettype id} = C_1\sigma_{1,1}\cdots\sigma_{1K_1}; \cdots ; C_n\sigma_{n,1}\cdots\sigma_{n,K_n} \text{ in } e : \tau}$$

$$\frac{(\text{lettype id} = C_1\sigma_{1,1}\cdots\sigma_{1K_1}; \cdots ; C_j\sigma_{j,1}\cdots\sigma_{jK_j}; \cdots ; C_n\sigma_{n,1}\cdots\sigma_{n,K_n}) \in \Delta \quad \Gamma;\Delta \triangleright M_1 : \sigma_{j,1} \quad \cdots \quad \Gamma;\Delta \triangleright M_{K_j} : \sigma_{j,K_j}}{\Gamma ; \Delta \triangleright C_j\; M_1 \ldots M_{K_j} : \text{id}}$$

$$\frac{\begin{array}{l}\Gamma;\Delta \triangleright M : \text{id} \quad (\text{lettype id} = C_1\sigma_{1,1}\cdots\sigma_{1,K_1}; \cdots ; C_n\sigma_{n,1}\cdots\sigma_{nK_n}) \in \Delta \\ \Gamma \cup \{x_{1,1} : \sigma_{1,1}, \ldots, x_{1K_1} : \sigma_{1,K_1}\}; \Delta \triangleright N_1 : \tau \ldots \\ \Gamma \cup \{x_{n,1} : \sigma_{n,1}, \ldots, x_{nK_n} : \sigma_{n,K_n}\}; \Delta \triangleright N_n : \tau\end{array}}{\Gamma;\Delta \triangleright \text{case } M \text{ of } \{C_1\, x_{11}\ldots x_{1K_1} \to N_1, \ldots, C_n\, x_{n1}\ldots x_{nK_n} \to N_n\} : \tau}$$

Mauricio Alfonso 65/09

2.b) Small-step

$$\frac{}{\text{lettype } id = C_1\, \sigma_{1,1} \ldots \sigma_{1,k_1}; \ldots ; C_n\, \sigma_{n,1} \ldots \sigma_{n,k_n} \text{ in } e \to e}$$

$$\frac{M_i \to M_i'}{C\ V_1 \ldots V_{i-1}\ M_i\ M_{i+1} \ldots M_n \to C\ V_1 \ldots V_{i-1}\ M_i'\ M_{i+1} \ldots M_n}$$

$$\frac{M \to M'}{\begin{array}{l}\text{case } M \text{ of } \{C_1\, x_{11} \ldots x_{1k_1} \to N_1, \ldots, C_n\, x_{n1} \ldots x_{nk_n}\} \to \\ \text{case } M' \text{ of } \{C_1\, x_{11} \ldots x_{1k_1} \to N_1, \ldots, C_n\, x_{n1} \ldots x_{nk_n}\}\end{array}}$$

$$\frac{}{\begin{array}{l}\text{case } C_i\ V_{i,1} \ldots V_{i,k_i} \text{ of } \{C_1\, x_{11} \ldots x_{1k_1} \to N_1, \ldots, C_i\, x_{i,1} \ldots x_{i,k_i} \to N_i, \ldots, C_n\, x_{n,1} \ldots x_{n,k_n} \to N_n\} \\ \to N_i\{x_{i1} \leftarrow V_{i,1}, \ldots, x_{ik_i} \leftarrow V_{i,k_i}\}\end{array}}$$

2.c) Big-step

$$\frac{}{\text{lettype } id = C_1\, \sigma_{1,1} \ldots \sigma_{1,k_1}; \ldots ; C_n\, \sigma_{n,1} \ldots \sigma_{n,k_n} \text{ in } e \Downarrow e}$$

$$\frac{M_1 \Downarrow V_1 \ \ldots \ M_n \Downarrow V_n}{C\, M_1 \ldots M_n \Downarrow C\, V_1 \ldots V_n}$$

$$\frac{M \Downarrow C_i\ V_{i,1} \ldots V_{i,k_i} \qquad N_i\{x_{i,1} \leftarrow V_{i,1}, \ldots, x_{i,k_i} \leftarrow V_{i,k_i}\} \Downarrow W}{\text{case } M \text{ of } \{C_1\, x_{11} \ldots x_{1k_1} \to N_1, \ldots, C_i\, x_{i1} \ldots x_{ik_i} \to N_i, \ldots, C_n\, x_{n1} \ldots x_{nk_n} \to N_n\} \Downarrow W}$$

Mauricio Alfonso 65/09

3) a) • $\forall X \forall Y ((b(x) \wedge \neg S(Y,Y)) \supset S(X,Y))$ (*1)

• $\neg \exists X \exists Y \; b(x) \wedge S(Y,Y) \wedge S(X,Y)$ (*2)

b) Paso (*1) a forma clausal:

$\forall x \forall Y ((b(x) \wedge \neg S(Y,Y)) \supset S(x,Y))$

$= \forall x \forall y \; \neg(b(x) \wedge \neg S(Y,Y)) \vee S(X,Y)$

$= \forall x \forall Y \; \neg b(x) \vee S(Y,Y) \vee S(X,Y)$

$= \{\neg b(X_1), S(Y_1,Y_1), S(X_1,Y_1)\}$ ①

Paso (*2) a forma clausal:

$\neg \exists X \exists Y \; b(x) \wedge S(Y,Y) \wedge S(X,Y) = \forall X \neg \exists Y \; b(x) \wedge S(Y,Y) \wedge S(X,Y)$

$= \forall X \forall Y \; \neg(b(x) \wedge S(Y,Y) \wedge S(X,Y)) = \forall X \forall Y \neg b(x) \vee \neg S(Y,Y) \vee \neg S(X,Y)$

$= \{\neg b(X_2), \neg S(Y_2,Y_2), \neg S(X_2,Y_2)\}$ ②

Quiero ver que ① $\wedge$ ② $\supset \neg \exists x \, b(x)$ ~~(*3)~~

~~Paso $\neg \exists x \, b(x)$ a forma clausal:~~

esto equivale a ver que ① y ② y $\neg(\neg \exists x \, b(x))$ es insatisfactible

$\neg(\neg \exists x \, b(x)) = \exists x \, b(x)$

preservo satisfac. con $b(c) = \{b(c)\}$ ③

Uso resolución entre ③ y ①

$$\frac{\{b(c)\} \quad \{\neg b(X_1), S(Y_1,Y_1), S(X_1,Y_1)\}}{\sigma(\{S(Y_1,Y_1), S(X_1,Y_1)\}) = \{S(Y_1,Y_1), S(c,Y_1)\}} \quad \sigma_1 = \{X_1 \leftarrow c\}$$ ④

resol. entre ④ y ②

$$\frac{\{S(Y_1,Y_1), S(c,Y_1)\} \quad \{\neg b(X_2), \neg S(Y_2,Y_2), \neg S(X_2,Y_2)\}}{\sigma_2(\{\neg b(X_2)\}) = \{\neg b(c)\}}$$ ⑤

$\sigma_2 = MGU(\{Y_1 \doteq Y_2, c \doteq X_2\})$

$\sigma_2 = \{X_2 \leftarrow c\}$

(cont.)

Scanned by CamScanner

Mauricio Alfonso 65/09

3) (cont.)

Resol. entre 5 y 3 $\dfrac{\{\neg b(c)\} \quad \{b(c)\}}{\sigma_3(\{\}) = \square}$ $\sigma_3 = \phi$

Luego no puede existir un barbero.